



Concours national d'informatique

Épreuve écrite d'algorithmique

ORGANISATEUR ORGANISEUR

1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre épreuve régionale. Sa durée est de 3 heures. Par la suite, une épreuve de programmation sur machine (3 heures 30).

Conseils

- Lisez bien tout le sujet avant de commencer.
- **Soignez la présentation** de votre copie.
- N'hésitez pas à poser des questions.
- Si vous avez fini en avance, relisez bien, ou préparez votre présentation pour l'entretien.
- N'oubliez pas de passer une bonne journée.

Remarques

- Le barème est donné à titre indicatif uniquement.
- Indiquez lisiblement vos nom et prénom, la ville où vous passez l'épreuve et la date en haut de votre copie.
- Tous les langages sont autorisés, veuillez néanmoins préciser celui que vous utilisez.
- Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe, sinon ça va barder.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le-nous, nous pouvons ajouter des questions plus difficiles.

2 Introduction

Ce sujet est réparti en 4 parties indépendantes :

- Puissance 4 (10 points)
- Vantardise (4 points)
- Régulateur d'expression (4 points)
- Flottille (17 points)

Si vous avez des difficultés sur une question, n'hésitez pas à passer à la suivante.

Aujourd'hui, il faut préparer et envoyer les lettres aux participants. C'est un projet en soi puisqu'on parle de milliards de lettres à faire¹.

Nous n'allons pas commencer maintenant à faire les enveloppes puisque seuls Joseph et Baptiste sont présents. Joseph s'ennuyant, il trouve une boîte de Puissance 4 !

Après une courte partie contre Baptiste, Joseph comprend que son opposant semble imbattable. En prétendant devoir finir un projet, le perdant s'en va en concevoir un programme pour être sûr de gagner ! Vous allez aider ce pauvre organisateur à battre Baptiste.

¹On n'est pas très sûr de nos calculs.

3 Puissance 4

3.1 Introduction

Le Puissance 4 est un jeu qui se joue sur une grille de 6 lignes et 7 colonnes.

2 joueurs s'opposent avec chacun des pions d'une couleur différente (ici "1" pour Joseph et "2" pour Baptiste).

Les joueurs posent un pion chacun leur tour. La grille étant verticale, les pions tombent sur la case vide la plus basse de la colonne. Le premier qui aligne (en ligne, en colonne ou en diagonale) 4 pions gagne.

Une partie peut être nulle si toutes les cases sont remplies sans qu'il n'y ait de gagnant.

Pour faire son programme, Joseph choisit de représenter la grille sous la forme d'un tableau en 2 dimensions de 6 lignes et 7 colonnes. La grille est à utiliser de cette manière : `grille[ligne, colonne]`.

Il choisit de représenter les cases avec des chiffres de la manière suivante :

- 0 une case vide,
- 1 une case avec un de ses pions,
- 2 une case avec un pion adverse

3.2 Fondations fondamentales

Question 1 (1 point)

Proposez une fonction qui, pour une colonne donnée, retourne la ligne de la case vide la plus basse, et -1 si la colonne est pleine.

La fonction prendra en paramètre la grille sous la forme d'un tableau en 2 dimensions ainsi que l'index de la colonne.

Question 2 (1 point)

Écrivez une fonction retournant les index des colonnes où il est possible de jouer (celles qui ne sont pas pleines).

Question 3 (2 points)

Décrivez un algorithme qui indique le numéro du gagnant sur une grille donnée. Si aucun joueur ne gagne, retournez -1. La grille ne peut contenir qu'un seul alignement de 4 pions.

Question 4 (1 point)

Si on utilisait l'algorithme que vous venez de faire à chaque tour, cela serait très coûteux sachant que de nombreux tests seraient faits *un trop grand nombre de fois*. Proposez une technique pour améliorer cet algorithme.

3.3 Minimisation maximale

L'algorithme Minimax est un algorithme récursif utilisé pour trouver le meilleur coup² dans un jeu de somme nulle³ sans aléatoire. Cela tombe bien, c'est le cas du Puissance 4 !

L'algorithme consiste à simuler toutes les parties possibles récursivement jusqu'à un certain nombre de tours (on appelle ça la profondeur) ou jusqu'à ce que la partie se termine (victoire ou nulle). Chaque récursion donne une situation. On attribue un score à chaque situation finale à l'aide d'une fonction d'évaluation. On remonte la récursion avec les valeurs vers la situation initiale en choisissant alternativement le meilleur et le pire score (on maximise le score lorsque c'est à nous de jouer, et on le minimise lorsque c'est à l'adversaire de jouer⁴).

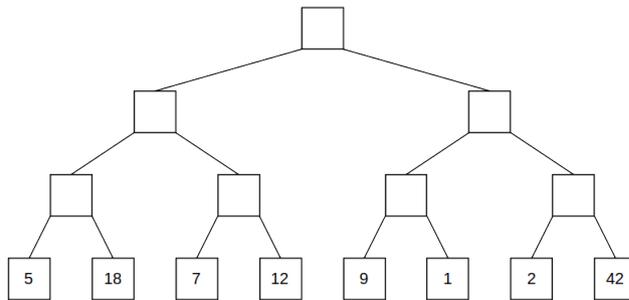


Figure 1: Exemple pour un jeu à deux coups possibles

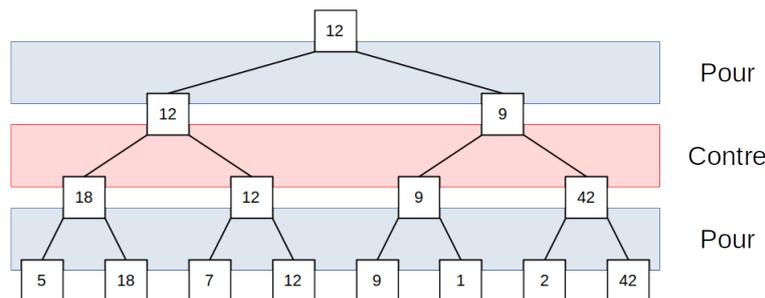


Figure 2: Le même exemple après la remontée

Nous évaluons chaque grille à l'aide d'un score qui représente à quel point la grille est en notre faveur. Lorsque 4 pions sont alignés, on ajoute une infinité de points au score. 3 pions alignés et 1 case vide ajoutent 5 points au score. 2 pions alignés et 2 cases vides ajoutent 2 points au score. 3 pions alignés pour l'adversaire et 1 case vide retire 5 points du score.

Question 5

(5 points)

Écrivez un algorithme Minmax pour aider Joseph à gagner.

Aidez-vous des indications précédemment données. Vous n'avez pas besoin d'implémenter la fonction d'évaluation de grille.

Si vous bloquez sur cette question, n'hésitez pas à passer à la partie suivante.

²L'algorithme peut théoriquement trouver le meilleur coup. En réalité, on ne poussera jamais la recherche jusqu'au bout, le coup trouvé sera dans ce cas-là un très bon coup et non le meilleur coup.

³Un jeu de somme nulle est un jeu où la somme des gains et des pertes de tous les joueurs est égale à 0. Cela signifie donc que le gain de l'un constitue obligatoirement une perte pour l'autre.

⁴d'où le nom Minmax ;)

4 Vantardise

Certains joueurs de puissance 4 sont mauvais gagnants⁵. Étant imbus d'eux-mêmes, ils ne se privent pas de se vanter de la moindre victoire. Notre but est de démasquer ces mauvais joueurs pour ne pas jouer contre eux. Baptiste souhaite automatiser la recherche des vantardises des mauvais gagnants⁶.

Question 6

(1 point)

Une pratique courante des vantards est de parler en *palindromes*⁷. Proposez un algorithme prenant en paramètre une chaîne de caractères et retournant Vrai si celle-ci est un palindrome, Faux sinon. (La casse n'a pas d'importance ici, les espaces et la ponctuation non plus).



Exemples de palindromes : stats - Engage le jeu que je le gagne. - Un emu a son os au menu. - Laval - L'ame sure ruse mal.

Question 7

(2 points)

Les mauvais joueurs utilisent souvent les mêmes mots dans leurs tirades. Nous voulons savoir le nombre d'occurrences d'un même mot dans un message. Écrivez un algorithme qui prend en paramètre deux chaînes de caractères. La première est le message et la deuxième le mot que nous cherchons. La fonction retourne le nombre d'occurrences (0 si aucune).

Question 8

(1 point)

Les vantards écrivent tellement vite qu'ils parenthèsent mal leurs messages. Écrivez une fonction qui retourne Faux si le message est bien parenthésé et Vrai sinon.



"Salut ((tout le monde) ! (Ça farte ?)" → Faux ; "Salut ((tout) le monde) ! (Ça farte ?)" → Vrai

⁵Tel que Joseph

⁶Tel que votre organisateur préféré

⁷Un palindrome est un mot ou une phrase dont l'ordre des lettres reste le même, qu'on les lise de gauche à droite ou de droite à gauche.

5 Régulateur d'expressions

Avant d'envoyer les lettres, il faut que l'on trouve les adresses postales où les envoyer. La "base de données" ressemble bien plus à rien qu'à une réelle base de données. Voici un extrait :

```
>Noel Flantier-181 chemin des rateaux ;Soirol , 12347 < Emilie Cousin - 17 avenue du  
General Seux , Le Lagron ,41927 < Lucienne Bramare- rue des Cerises , Daiville ,19687  
<
```

Comme vous le voyez avec ces 3 adresses, la syntaxe est aléatoire.

Utiliser une fonction qui extrait les informations serait très long à coder. Alors, il nous faut donc utiliser des expressions régulières afin d'extraire toutes les informations automatiquement.

5.1 Expression régulière

Une expression régulière est une chaîne de caractère (aussi appelé motif) qui décrit un ensemble de chaîne de caractères.

Pour ce faire, nous allons commencer par utiliser 4 quantificateurs :

- | sépare des groupes alternatifs : `Juli(e|1|oz)` décrit "Julie", "Juli1" et "Julioz".
- ? définit un groupe qui existe zéro ou une fois : `Prol?ogin` décrit "Proogin" et "Prologin".
- + définit un groupe qui existe au moins une fois : `(Pou)+` décrit "Pou", "PouPou", "PouPouPou" et ainsi de suite.
- * définit un groupe qui existe zéro, une ou plusieurs fois : `Idotno*` décrit "Idotn", "Idotno", "Idotnoo" et ainsi de suite.

Remarquez l'importance des parenthèses qui permettent d'appliquer un quantificateur à plusieurs caractères.

Par exemple, l'expression régulière `ch(a|io)t(on)?` reconnaît les expressions : "chat", "chaton", "chiot" et "chioton" ⁸.

Autre exemple, `Uu*bun(tu)+` reconnaît les expressions : "Ubuntu", "Uubuntu", "Uubuntutu", "Uubuntutu", "Uuuuubuntu", "Uuubuntututututu", et une infinité d'autres. Cependant, "Uuubun", "Uubuntuu" ou "NixOs" ne sont pas reconnus.

Il y a parfois plusieurs manières de décrire un ensemble de chaînes de caractères.

Par exemple : `(aa)|(ab)` reconnaît le même ensemble que `aa|b`.

5.2 Motifation

Question 9 (1 point)

Trouvez les expressions régulières plus courtes aux motifs `(ab)|a` et `aa*`.

Question 10 (1 point)

Utilisons le sucre syntaxique⁹ `[a-z]` qui ici décrit toutes les lettres de l'alphabet minuscules dans l'intervalle de "a" à "z", il permet de simplifier l'écriture du motif `a|b|c|...|y|z`.
Écrivez une expression régulière qui reconnaît les prénoms.
Un prénom commence avec une lettre majuscule et est suivi de 0 à une infinité de lettres minuscules.
On ignore les accents et les tirets.

Question 11 (2 points)

Écrivez un motif qui reconnaît une adresse postale de notre base de données.
Pour montrer un espace, nous allons utiliser le caractère "sous-tiret" : "_".

⁸Au cas où vous ayez un doute, ce mot n'existe pas.

⁹Le sucre syntaxique ne se mange pas. C'est une expression simplifiant une ou plusieurs expressions.

6 Flottille

6.1 Réseau de flot

Un réseau de flot est un graphe orienté¹⁰ où les arcs (des arêtes orientées) sont pondérés avec une capacité. Cette capacité indique le flot maximal que peut accepter un arc.

Le flot est un débit entre la source et le puits.

Pour qu'un flot soit valide, il faut que la somme des flots entrants à un nœud, soit égale à la somme des flots sortants de ce même nœud, sauf pour la source et le puits. La source est le nœud d'où sortent les flots et le puits celui où rentrent les flots.

La source et le puits doivent être deux nœuds distincts.

On représente le flot et la capacité d'un arc de cette manière : *flot/capacité*.

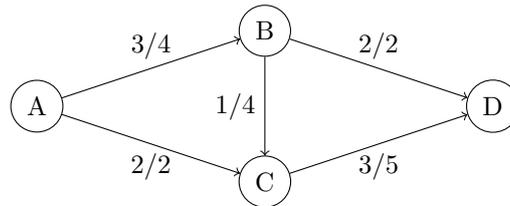


Figure 3: Exemple simple de flot où A est la source et D le puits

6.2 Envoi des enveloppes

Il y a 2 grandes traditions à Prologon, la finale et l'envoi des enveloppes. Intéressons-nous à la seconde !

L'objectif de cette journée est de faire toutes les enveloppes pour les candidats et les écoles. Il y a plusieurs étapes :

1. Faire les paquets avec 2 affiches et la lettre. Un organisateur peut faire 10 paquets par minute.
2. Entrer chaque paquet dans une enveloppe. Une personne arrive à mettre 20 paquets par minute.
3. Vérifier chaque enveloppe. Tout le monde peut vérifier 40 enveloppes par minute.
4. Fermer chaque enveloppe. Quelqu'un d'entraîné ferme 30 enveloppes par minute.

On se rend bien compte que les étapes n'ont pas toutes le même débit. Nous allons chercher à optimiser le processus.

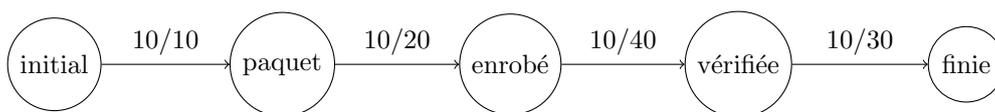


Figure 4: Processus avec seulement 4 organisateurs : HéRisson, Amélie, Valérian et Clément

On peut ici se rendre compte que l'on peut faire que 10 lettres par minute, puisque la confection des paquets limite.

Question 12

(2 points)

Romain vient d'arriver, indiquez sa place pour augmenter le débit. Maintenant Elsa, Raphaël, Ethan et Oscar viennent en renfort, où faudrait-il les mettre et pour maximiser le débit. Quel est ce débit ?

¹⁰Le truc avec les nœuds et les arêtes

Question 13

(3 points)

Proposez un algorithme qui indique la manière dont il faut répartir un certain nombre d'organisateur afin de maximiser la production d'enveloppes.

Celle-ci prend en paramètre le nombre d'organisateur par étape sous la forme d'un tableau d'entier ainsi que le nombre d'organisateur à répartir. L'algorithme doit renvoyer la répartition sous la forme d'un tableau d'entier.

6.3 Tu peux me passer l'eau ?

L'organisateur qui écrit ce sujet s'est rendu compte que le problème était plus simple à résoudre que prévu. Le bureau de Prologin s'en excuse.

Maintenant que nous avons parlé de la plus grande tradition de l'association, parlons du rituel journalier. Tous les jours, il y a tout le temps quelqu'un qui demande "Tu peux me passer l'eau ?" alors que la carafe est de l'autre côté de la salle.

Le plus gros problème est qu'une fois que les milliards de lettres sont faites, le local devient très encombré. À cause de l'encombrement, la personne assoiffée ne peut pas se lever¹¹. C'est donc une chaîne humaine qui se met en place.

Cette fois-ci, Valérian demande à ce qu'on lui passe un bonbon, puis un autre, puis un autre, puis un autre...

Ici Oscar (O) sera la source et Valérian (V) le puits¹². La capacité d'un arc est le nombre maximum de bonbons que deux organisateurs peuvent se passer par minute avant que l'un deux s'inquiète de la dentition de Valérian.

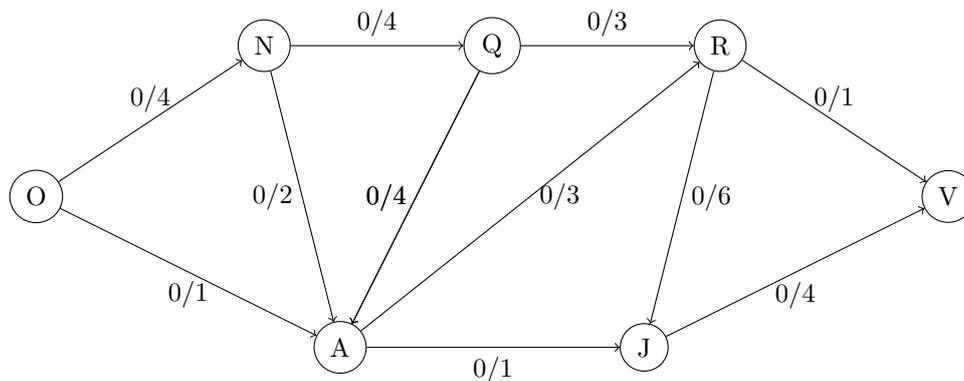


Figure 5: Réseau représentant les occupants

Question 14

(2 points)

Le problème du flot maximum consiste à trouver le flot maximal valide dans un réseau. Un flot est maximal s'il n'existe pas un autre flot valide qui a un débit strictement supérieure.

Dessinez le flot maximum de ce graphe.

Question 15

(4 points)

Proposez un algo trouvant le flot maximum de tous les graphes.

On vous donnera le graphe (dans la structure que vous souhaitez (précisez-la)), la source et le puits.

Question 16

(6 points)

Proposez un algorithme qui, à partir d'un graphe orienté, retourne les chemins qu'il faut pour parcourir tous les nœuds. Le nombre de chemins doit être minimal. Deux chemins différents ne peuvent utiliser un même arc. Pour résoudre ce problème, utilisez des réseaux de flot. Vous êtes libre de l'implémentation du graphe, précisez-la.

¹¹Pas seulement par paresse.

¹²Toute personne appelant ce vénérable membre "le puits" sera exclu à vie du concours, ou quelque chose comme ça.

7 Bonus

Question bonus 17 (1 point)

▮ Proposez un titre d'un seul-en-scène que vous pourriez proposer lors de la finale.

Question bonus 18 (3 points)

▮ Écrivez une blague de ce spectacle.

Question bonus 19 (2 points)

▮ Dessinez votre meilleur mème sur Prologin.¹³

Question bonus 20 (1 point)

▮ Est-ce que le sujet est bien parenthésé ?

¹³Vous pouvez le poster sur r/prologin